

SPECIFICATION

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN, that Joel Ronning, a resident of Excelsior, Minnesota and a citizen of the United States; Kelly Wical, a resident of Hastings, Minnesota and a citizen of the United States; and Marc Kukura, a resident of Minneapolis, Minnesota and a citizen of Canada, have invented certain new and useful improvements in

APPARATUS AND METHOD FOR CONTROLLING

DISTRIBUTION OF FILES AND FILE UPDATES

of which the following is a specification.

**APPARATUS AND METHOD FOR CONTROLLING
DISTRIBUTION OF FILES AND FILE UPDATES**

REFERENCE TO RELATED APPLICATIONS

5 The present application is a continuation-in-part of United States Patent Application
Serial No. 09/372,253, filed August 11, 1999, and entitled "Apparatus and Method for Adaptive
Fraud Screening for Electronic Commerce Transactions," which is a continuation-in-part of
provisional United States Patent Application Serial No. 60/110,952, filed December 4, 1998, and
entitled "Apparatus and Method for Providing Electronic Commerce," both of which are
incorporated herein by reference as if fully set forth.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000

 The present application is related to the following applications, all of which are
incorporated herein by reference as if fully set forth: United States Patent Application of Joel
Ronning, Kelly Wical, and Marc Kukura, entitled "Apparatus and Method for Controlling
Distribution of Files and File Updates," and filed on even date herewith; United States Patent
Application of Joel Ronning and Kelly Wical, entitled "Apparatus and Method for Secure
Downloading of Files," and filed on even date herewith; United States Patent Application of Joel
Ronning and Kelly Wical, entitled "Cache Management for Dynamic Web Pages," and filed on
even date herewith; and United States Patent Application of Joel Ronning and Kelly Wical,
entitled "Global Web Site Management," and filed on even date herewith.

FIELD OF THE INVENTION

 The present invention relates to an apparatus and method for controlling distribution of
files and updates to files.

BACKGROUND OF THE INVENTION

Various systems exist for distribution of software or files over the Internet or other networks. With these systems, a user requests to download a particular software file, and the corresponding system transmits the file to the user's machine. The transmission may occur using conventional protocols for distribution of information over the Internet. If the transmission becomes interrupted due to, for example, loss of the network connection, the user must manually request to download the file again, and typically must attempt to download the entire file. Some particular systems exist for transmitting only a portion of a file. However, the systems are often specific to a particular network and require manual interaction by the user as well.

In addition, after downloading the software file, a user is not necessarily informed of any updates to the file. Software is frequently updated in order to provide a new version of it and add additional features or change features based on user feedback. However, a user who downloads the software file does not necessarily receive any updates of the software unless the user specifically and manually requests it. Upon requesting an update, a server does not necessarily know the exact version of the software file that the user has and thus does not necessarily know which particular update to provide for that file.

Accordingly, a need exists for improved systems for distribution of digital information and for providing updates to files.

SUMMARY OF THE INVENTION

A method and apparatus consistent with the present invention use an agent software application for controlling distribution of files and managing updates to files. The agent

automatically downloads files upon request by a user and can download a file in multiple portions by tracking an amount of a received portion of the file.

The method and apparatus include transmitting a request to download a file, the request including an identification of the file and an indication of starting point for transmission of the file. A serial transmission of digital information for the file is received beginning at the starting point, and the digital information is appended to a previously-received portion of the file, if present. The received digital information is also stored.

The agent can also search for updates to files by using application signatures to uniquely identify files stored on a user's machine and transmitting the application signatures to a server storing the updates. In addition, the agent uses a number of screens for displaying information to a user concerning the downloading of files and updates to files. The screens also permit the user to enter schedule information to control downloading of files and searching for updates to files.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an environment for providing electronic commerce.

FIG. 2 is a block diagram of an electronic commerce system.

FIG. 3 is a block diagram of exemplary hardware components of an electronic commerce system.

FIG. 4 is a flow chart of processing for an electronic commerce system.

FIGS. 5a and 5b are a flow chart of a home screen routine for implementing an agent to manage distribution of digital information and file updates.

FIG. 6 is a flow chart of a downloads screen routine for the agent.

FIG. 7 is a flow chart of a downloads routine for distribution of digital information using the agent.

FIG. 8 is a flow chart of an updates screen routine for the agent.

FIGS. 9a and 9b are a flow chart of find updates routine for the agent to locate and download information concerning update to files.

FIGS. 10-24 are diagrams of exemplary screens supported by the routines shown in FIGS. 5-9 and used to implement an embodiment of an agent for distribution of digital information and file updates.

FIG. 10 is a diagram of a home screen.

FIG. 11 is a diagram of a news and deals screen.

FIG. 12 is a diagram of a shopping screen.

FIG. 13 is a diagram of a customer service screen.

FIG. 14 is a diagram of a privacy screen.

FIG. 15 is a diagram of a general settings screen.

FIG. 16 is a diagram of a tour screen.

FIG. 17 is a diagram of a downloads screen.

FIG. 18 is a diagram of an install screen.

FIG. 19 is a diagram of a schedule downloads screen.

FIG. 20 is a diagram of a download help screen.

FIG. 21A is a diagram of a download status screen.

FIG. 21B is a diagram of a continue download screen.

FIG. 22 is a diagram of an updates screen

FIG. 23 is a diagram of an updates help screen.

FIG. 24 is a diagram of an update status screen.

DETAILED DESCRIPTION

Overview

5 As shown in FIG. 1, an end user at computer 100 enters a particular client web site 102 through connection 101 in order to shop for products, which may include, but is not necessarily limited to, digital products. Digital products include any information capable of being represented in digital form; examples of digital products include, but are not limited to, the following: software applications; data, music, video, books, text, audio-visual information, multimedia information, graphical information, fonts, and artistic works.

10 The end user may view a page from the client's web site, for example, listing and describing various products. When the end user indicates a desire to purchase a product by, for example, selecting a purchase icon on the client's web page, the end user's connection 103 is transferred to a commerce network server providing the client's on-line web store 104, and this typically occurs as a background process. The end user may make a secure purchase 105 through page 106 from a product library and, as represented by arrow 107, the purchased product is delivered electronically over the network connection, physically such as by mail, or both electronically and physically.

15 The purchase typically involves the end user entering payment and related information, such as a credit card number and an associated name and address. In response, the commerce network server determines whether to accept the order and, if accepted, may provide a secure download of the purchased product to the end user's computer for digital products. Determining

whether to accept or decline the order involves the use of accessing information concerning prior attempted purchases using information related to or associated with the information in the order, and determining from the related or associated information the likelihood that the order involves a fraudulent attempt to obtain products. Although only one computer 100 and web site 102 are shown, a commerce network server may provide multiple on-line web stores 104 and may interact with end users at multiple computers and multiple web sites.

Electronic Commerce System and Process

FIG. 2 is a block diagram of an electronic commerce system 200 illustrating interaction between an end user machine 201 and a server 205, illustrating exemplary software modules, caches, and related databases. Server 205 may correspond with the commerce network server described above providing on-line web stores. As shown, end user machine 201 interacts through network 204, such as the Internet or other type of network, with server 205. End user machine 201 may also access a web page on an intermediate server and subsequently be transferred to server 205. End user machine 201 may interact directly with server 205 or interact through an agent module 203, which performs processing specific to a user at end user machine 201. End user machine 201 transmits a request 228 or 229 to server 205 from agent module 203 or page 202, possibly including a request for a particular web page, a request to purchase and download a digital product, or a request for a search for a particular product. Although only one end user machine 201 is shown, server 205 may interact via network 204 with multiple end user machines and multiple intermediate servers maintaining web sites accessed by users at the end user machines. Other examples of systems providing electronic commerce are disclosed in the

following United States patents, all of which are incorporated herein by reference as if fully set forth: 5,907,617; 5,903,647; 5,887,060; 5,883,955; 5,883,954; and 5,870,543.

In server 205, the request may be transmitted through a uniform resource locator (URL) spoofer module 206, which performs initial processing of the URL. In particular, URL spoofer module 206 changes the URL so that if a user subsequently downloads a product, end user machine 201 by default saves the product under a file name associated with the product but gives no direct access to the physical location on the server. It may also misdirect the user in order to detect attempts to fraudulently obtain products. Alternatively, the request may be transmitted directly to a web server module 207, which performs initial processing on the request.

A log in module 208 receives the request and records certain data associated with the request, such as the user's request, Internet Protocol (IP) address, date and time, and particular demographic information. The request is then transmitted to a security module 209, which uses heuristics and other techniques in order to detect a person attempting to bypass particular steps of the process, or otherwise receive or access the products without providing payment.

A process request module 211 first checks a page/product cache 218 to determine if the requested web page has been previously requested or, if applicable, the relevant product has been previously requested. If so, process request module 211 accesses information in page/product cache 218 in order to avoid repeatedly generating the same information for the same or a similar request. If applicable, process request module 211 also checks a search cache 217 to determine if the requested search has been previously requested and, if so, it uses information in search cache 217 to generate particular output. A build cache module 212 within process request module 211, if applicable, builds information for storage in either of the caches.

A fraud (payment) module 210, typically within process request module 211, performs processing necessary to conduct the payment transaction, including processing of credit card information. It also records payment-related information.

A build output module 213 next assembles information for the request. It first checks a graphical user interface (GUI) implementation cache 216 to determine if a requested web page has been previously constructed and provided. If so, it may use the information in GUI implementation cache 216 to avoid unnecessary repeated processing of the same information. If applicable, a build cache module 214 within build output module 213 creates information for storage in GUI implementation cache 216.

A log completion module 215 performs final processing on the request. If the request is only for a web page or search, log completion module 215 transmits the web page or search information back to end user machine 201. If end user machine 201 uses agent module 203, log completion module 215 may transmit information to an agent transaction server 219, which constructs and creates the web page based upon that information and transfers it back to agent module 203 for construction of the web page at page 202 on end user machine 201.

If the request included a request for a product, log completion module 215 transmits the request to a download processor 224, which checks with process request module 211 to verify the authenticity of the request and perform a security check. If the request is valid, as determined by information transmitted back from process request module 211, download processor 224 securely transmits the requested product from a product download database 225 to end user machine 201. The transmitted product may be transmitted through agent transaction server 219 if end user machine 201 uses agent module 203.

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450 460 470 480 490 500 510 520 530 540 550 560 570 580 590 600 610 620 630 640 650 660 670 680 690 700 710 720 730 740 750 760 770 780 790 800 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950 960 970 980 990 1000

The following provides a description of each database shown in FIG. 2. A data warehouse database 221 provides log in information along with keys, which provides an index to associated information in a commerce database 223. Commerce database 223 contains data tables storing information related to products and requests, such as a product table, order table, and other such tables. A summary database 220 provides information from the data warehouse database in summary form. Product download database 225 provides products in digital form for retrieval by the download processor. This configuration provides the advantage of storing the products without wrappers or associated passwords, and instead providing for a secure download of the products. Wrappers or associated passwords may still be used, if desired.

A site GUI/feature control database 226 and default GUI/feature control database 227 may be accessed by process request module 211 and build output module 213 for storage and retrieval of information related to web sites.

A credit card fraud database 222 stores credit card transaction information, including credit card processing history, and other information for adaptive fraud screening. Such information may thus be used in performing another security check. For example, the database may store a list of stolen credit card numbers.

FIG. 3 depicts an exemplary data processing system 300 with a computer 301 illustrating exemplary hardware components of end user machine 201, server 205, and an intermediate server, if used to transfer the end user's connection. Computer 301 includes a connection with a network 307 such as the Internet or other type of network, which may correspond with network 204. Computer 301 typically includes a memory 302, a secondary storage device 304, a processor 305, an input device 306, a display device 303, and an output device 308.

Memory 302 may include random access memory (RAM) or similar types of memory, and it may store one or more applications 309 for execution by processor 305. Applications 309 may correspond with the modules shown in FIG. 2. Secondary storage device 304 may include a hard disk drive, floppy disk drive, CD-ROM drive, or other types of non-volatile data storage, and it may correspond with the various databases shown in FIG. 2. Processor 305 may execute applications or programs stored in memory 302 or secondary storage 304, or received from the Internet or other network 307. Input device 306 may include any device for entering information into computer 301, such as a keyboard, cursor-control device, or touch-screen. Display device 303 may include any type of device for presenting visual information such as, for example, a computer monitor or flat-screen display. Output device 308 may include any type of device for presenting a hard copy of information, such as a printer, and other types of output devices include speakers or any device for providing information in audio form.

Although computer 301 is depicted with various components, one skilled in the art will appreciate that this computer can contain additional or different components. In addition, although aspects of an implementation consistent with the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on or read from other types of computer program products or computer-readable media, such as secondary storage devices, including hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other network; or other forms of RAM or ROM. The computer-readable media may include instructions for controlling a computer system, such as computer 301, to perform a particular method.

FIG. 4 illustrates high-level processing 400 through electronic commerce system 200. In process 400, web server module 207 receives a request from end user machine 201 (step 401). Log-in module 208 processes the request to obtain log information (step 402). Security module 209 performs a security check (step 403). Process request module 211 checks page/product cache 218 (step 405) and checks search cache 217, if applicable (step 406). Fraud module 210 performs payment-related functions and process request module 211 processes the request and uses build cache module 212 to build a cache, if applicable (step 407). Build output module 213 checks GUI implementation cache 216 (step 408), assembles information for responding to the request, and uses build cache module 214 to build a cache, if applicable (step 409).

Log completion module 215 performs final processing of the request (step 410), and it determines if the request is a purchase request (step 411). If so, it transfers the request to download processor 224 (step 412), which securely downloads the requested product and delivers it to the end user machine 201 (step 413). If the request was not a purchase request as determined by step 411, log completion module 215 delivers page and optional search information to end user machine 201 (step 414).

Commerce System

This feature of electronic commerce system 200 involves providing a single transaction point for processing of electronic commerce, generally involving the series of modules shown in FIG. 2. Examples of the processing in this single transaction point include security, checking for fraud, on-line reporting, and processing orders. The benefits of providing such processing through a single transaction point typically include, for example, scalability, reliability of

processing, simplicity of design by avoiding multiple disparate systems, and avoiding repeated processing.

Fraud Detection and Prevention

5 This feature of electronic commerce system 200 involves preventing fraud and ensuring security during electronic commerce transactions. An electronic commerce transaction involves a user electronically requesting purchase of a product and providing payment for the product. The user may receive the product electronically for digital products or may receive it by other means such as via mail service. The terms fraud and fraudulent refer to attempts by a user to obtain a product without providing proper payment, or otherwise not satisfying requirements for obtaining a product. For example, a user may attempt to use a stolen or false credit card number to obtain a product or attempt to tamper with the system so that the user obtains a product for less than the required price. As another example, certain information related to a user's order may tend to indicate that the user is likely to fail to provide payment, even if the credit card number used appears valid, and system 200 attempts to make that determination based on an analysis of the user's information and associated information.

20 The fraud detection and prevention may also involve preventing users or customers from having access to certain stored data such as credit card information and products. It involves preventing users from bypassing particular modules or processing in system 200. Users typically interact with system 200 in a known manner for non-fraudulent transactions. Likewise, particular types of interaction tend to indicate that a user is attempting to circumvent system 200 and fraudulently obtain products. Therefore, system 200 analyzes a user's interaction, as well as a user's information submitted on an order typically using an order form and related information

in a database, in order to determine a likelihood or probability that the user is engaging in attempted fraud during the transaction. That determination is used as a basis to either accept or decline the order.

The fraud detection and prevention features of system 200 generally include functions within the security and fraud (payment) modules 209 and 210, as further explained below. System 200 may implement modules 209 and 210, or a single module or any number of modules to implement the features, using software stored in memory 302 or secondary storage 304 for execution by processor 305. Modules 209 and 210 may also be implemented using program products or a combination of software and specialized hardware components. In addition, the modules may be executed on multiple processors for processing a large number of transactions, if necessary or desired.

An exemplary implementation of modules 209 and 210 for fraud detection and prevention is disclosed in U.S. Application Serial No. 09/372,253, filed August 11, 1999, and entitled "Apparatus and Method for Adaptive Fraud Screening for Electronic Commerce Transactions."

Global Web Site Management

This feature of electronic commerce system 200 involves providing centralized management of host sites, managing all host sites through a central database. It may include, for example, extending a look and feel of a particular web site into another web site. For example, when an intermediate server transfers an end user's connection to a commerce network server, as described with respect to FIG. 1, the commerce network server may transmit a page having the same look and feel as the page on the intermediate server, thus providing an apparent seamless transition to the end user.

This feature also may include making local changes to a large number of managed web sites. Therefore, instead of making similar changes to each individual web site, system 200 may broadcast those particular changes and make the corresponding changes to the managed web sites. It thus provides an advantage, for example, of easily making changes to a large number of web sites. For example, it may automatically broadcast a few particular features every week to the managed web sites in order to regularly update the sites.

Security Area

This feature of electronic commerce system 200 provides for securely storing clients' products and providing a secure download process, typically without the use of wrappers or passwords. This feature generally involving the series of modules shown in FIG. 2. It includes a database type of security intended at least to simplify the purchase process for a user. A user typically need only enter a credit card number and in response receives a requested product.

Cache Management for Dynamic Web Pages

This feature of electronic commerce system 200 concerns a dynamic cache providing for a fast page response and dynamic pages, typically guaranteeing that pages contain current information. It generally involves caches 216, 217 and 218 shown in FIG. 2, providing cache processing for three areas: product searching; GUI implementation (particular features on a web page); and static page building (a web page as a whole).

The complexity of product searching often affects speed of a response. Providing caching of information means that the same information need not be repeatedly retrieved from an external database and constructed into a responsive web page. Rather, the information may be cached

locally and quickly retrieved in response to the same or a similar request. This feature maintains up-to-date information by knowing to create and destroy information interactively for the caches.

Intelligent Agent

5 An intelligent agent includes an application that resides locally on a client machine in order to perform processing specific to a user of the machine, generally involving agent module 203 and agent transaction server 219 shown in FIG. 2. It includes at least two aspects, a pull side and a push side. The pull side involves the agent obtaining information and knowing how to link to a server and provide the information to the user in a personal way, customized for that user. It shields the user from the order process, for example, in order to simplify it. The agent may also perform system management, for example, performing a background process that scans the user's system in order to manage licensed software, perform archival control, and perform other such processing.

The push side involves making special information available to the user through the agent. For example, if the user already ordered a particular product, the agent may inform the user of any bugs in the product, product upgrades, or related products. That information is "pushed" to the agent from the server. The server may provide initial filtering of information given to the agents, and the agents perform additional filtering in order to present the information in a specific way to the user.

20 Therefore, an agent manages distribution and download of files including digital products to a user's machine, as well as locating and downloading updates to files contained on the user's machine. The term "file" refers to any type of digital product. A file may include associated information concerning the file, such as a name and size, which is possibly useful for

downloading purposes. Alternatively, a file may include only a digital product itself or a portion of a digital product.

The agent can thus assist a user in managing the downloading of particular digital information, as well as provide recommendations concerning updates to files on the user's machine. An embodiment consistent with the agent may be implemented in software or program products within agent module 203 in user machine 201, and it may be included within agent modules on many particular end user machines interacting with server 205. FIGS. 5-9 are flow charts of routines for implementing an agent in agent module 203. The term "agent" refers to any software or other type of application for executing the functions in FIGS 5-9.

FIGS. 10-24 are diagrams of screens supported by the routines in FIGS. 5-9. The term "screen" refers to any visual element or combinations of visual elements for displaying information; examples include, but are not limited to, user interfaces on a display device or information displayed in web pages or in windows on a display device. The screens may be formatted, for example, as web pages in HyperText Markup Language (HTML), or in any other suitable form for presentation on a display device depending upon applications used by users to interact with the system.

The screens include various sections, as explained below, to provide information or to receive information or commands. The term "section" with respect to screens refers to a particular portion of a screen, possibly including the entire screen. Sections are selected, for example, to enter information or commands or to retrieve information or access other screens. The selection may occur, for example, by using a cursor-control device to "click on" or "double click on" the section; alternatively, sections may be selected by entering a series of key strokes or

in other ways such as through voice commands or use of a touch screen. In addition, although the screens shown in FIGS. 10-24 illustrate a particular arrangement and number of sections in each screen, other arrangements are possible and different numbers of sections in the screens may be used to accomplish the same or similar functions of displaying information and receiving information or commands. Also, the same section may be used for performing a number of functions, such as both displaying information and receiving a command.

FIGS. 5a and 5b are a flow chart of a home screen routine 500 for the agent. Routine 500 supports a home screen 700 shown in FIG. 10. Home screen 700 is a main screen shown to a user upon launching the agent. In routine 500, the agent typically resides active as a background process on a user's machine (step 502). By remaining active in the background, the agent may automatically execute certain functions, as scheduled by the user and further explained below. The user launches the agent, as determined by step 504. The user may launch the agent in a number of ways such as, for example, selecting the agent from a start menu, executing a run command, or selecting the agent as represented by an icon in a system tray on the user's computer. If the user launches the agent, the agent displays home screen 700 (step 506). The term "eBot" is a trademark of Digital River, Inc.

Home screen 700 includes a section 701 for use in providing information to the user and receiving information from the user. From home screen 700, a user may select a function by selecting a particular section of home screen 700 shown in FIG. 10 (step 508). Selection of home section 704 (step 510) causes display of home screen 700 (step 530). Selection of news and deals section 706 (step 512) causes display of a news and deals screen 726 shown in FIG. 11 (step 532). News and deals screen 726 illustrates an example of providing particular information

in section 701 to users. News and deals screen 726 typically includes a section 727 by which a user may connect with a particular web site as related to the information within section 701. The agent determines if the user selects section 727 (step 552); if so, the agent launches a web browser on the user's machine and connects with the appropriate web site (step 554). The agent may work with any type of web browser or other application executed by user machine 201 for network or Internet communication; examples of web browsers include the Netscape Navigator program and the Microsoft Internet Explorer program.

Selection in home screen 700 of a shopping section 708 (step 513) causes display of a shopping screen 728 shown in FIG. 12 (step 534). Shopping screen 728 permits display of information related to on-line shopping within section 701. In addition, shopping screen 728 typically includes a connect section 730 by which a user may connect with a particular web site. From shopping screen 728, the agent determines if the user selects connect section 730 (step 552); if so, the agent launches the web browser and connects with the appropriate web site for on-line shopping (step 554). The agent may store and associate connect section 730 with a URL or a uniform resource indicator (URI) for use in accessing a particular web site.

In home screen 700, selection of a customer service section 710 (step 514) causes display of a customer service screen 732 shown in FIG. 13 (step 536). Customer service screen 732 provides for display of information to a user concerning operation of the agent. It typically includes a plurality of connect sections 734 for a selection by a user in order to connect with an appropriate web site in order to obtain customer service information. Therefore, from customer service screen 732 the agent determines if a user selects one of the connect sections 734 (step

552); if so, the agent launches a web browser and connects with an appropriate web site for customer service information (step 554).

From home screen 700, selection of a privacy section 712 (step 516) causes display of a privacy screen 736 shown in FIG. 14 (step 538). Privacy screen 736 permits display of information to a user concerning the privacy of the information used by the agent for downloading information and providing for file updates. Privacy screen 736 also typically includes connect section 730 for connection with a web site to obtain information concerning privacy. From privacy screen 736 the agent determines if the user selects section 730 (step 552), and if so, the agent launches a web browser and connects with an appropriate web site for providing privacy information to a user (step 554).

From home screen 700, selection of a general settings section 714 (step 518) causes display of general settings screen 738 shown in FIG. 15 (step 540). From general settings screen 738, the agent receives and saves settings as entered by the user, including schedule information for finding updates (step 556). In particular, general settings screen 738 includes a name section 739 in which a user may enter a name or other identifier. Selection of a section 740 permits the user to instruct the agent to search for updates on a periodic basis. In particular, the user may enter a number of days in section 741 by which they want the agent to automatically search for update to the files on the user's machine, as explained below. Using the information in section 741, the agent automatically searches for updates to the files on the user's machine using a connection with server 205 based upon the days parameter in section 741. Instead of using a number of days parameter, the agent may alternatively search for updates on a different periodic basis such as the first day of every month, or at a particular time and day each week or month.

Selection of a section 742 permits the user to specify a particular proxy server for downloads as entered in section 743 identifying a server and port for use with the web browser. Selection of a section 744 permits the user to specify automatic downloads such that the agent will automatically download a file without prompting the user for permission. Selection of a section 745 permits the user to instruct the agent to download from any web site on the Internet through server 205. Selection of a section 746 permits the user to instruct that the agent software be activated upon system start-up of the user's machine.

From home screen 700, a selection of a tour section 716 (step 520) causes display of a tour screen 748 shown in FIG. 16 (step 542). Tour screen 748 permits the agent to display information concerning its operation to the user. The user may typically obtain more information concerning operation of the agent by selecting connect section 730 in order to connect with an appropriate web site. Therefore, on tour screen 748 the agent determines if the user selects connect section 730 (step 552); if so, the agent launches a web browser and connects with an appropriate web site in order to provide information concerning its operation (step 554).

From home screen 700, selection of a downloads section 718 (step 552) causes execution of a downloads screen routine (step 554), as further explained below. Selection of an updates section 720 in home screen 700 (step 524) causes execution of an update screen routine (step 546), as further explained below. A banner section 722 in home screen 700 permits the agent to display advertising or other information to a user. The user may select the banner in order to connect with a corresponding web site and obtain more information or shop on-line for products or services. Therefore, a selection of banner section 722 (step 526) causes the agent to launch a web browser and connect with an appropriate web site as determined by network address

information, such as a URL, associated with the banner (step 548). If the user selects a close section 724 (step 528), the agent closes home screen 700 (step 550).

The agent may also be programmed or instructed by the user to automatically perform downloads of files and search for file updates using time parameter information. The agent monitors a current date and time, and compares it with schedule information to detect occurrence of a particular date and time (step 557). The agent can monitor a current date and time by accessing an internal calendar and clock for user machine 201 on which it runs.

The agent determines if the user has scheduled a particular function (step 558). That determination may be based upon comparing stored time parameter information as entered by a user with time information as determined by settings on a user's machine, as monitored in step 557. If the user has scheduled a particular download for this time (step 560), the agent executes a downloads routine based on the scheduled information (step 562). If the user has scheduled an update for this time (step 564), the agent executes a find updates routine based upon the scheduled information (step 566). In order for the scheduled function to occur, the agent software typically must be active as a background process such that it is operational and the user's machine must maintain or provide for a network connection such that the agent may automatically perform the downloads and search for the file updates through the network connection without requiring further interaction by the user.

FIG. 6 is a flow chart of downloads screen routine 544 activated upon the user's selection of downloads section 718 in home screen 700. Upon selection of downloads section 718, the agent displays a downloads screen 750 shown in FIG. 17 (step 568). Downloads screen 750 includes a number of sections for providing status information to a user concerning downloads

and for a user to instruct the agent to perform certain operations concerning downloads. In routine 544, a user selects a function within downloads screen 750 (step 570), and the agent in response executes the selected function. Selection of a download section 752 causes execution of a downloads routine (step 584), as further explained below. For the download information, downloads screen 750 provides a status section 764 having status information concerning files. In particular, a name column 766 provides a listing of file names, a status column 768 provides identification on the status of each file, a completion column 770 provides information concerning an amount of the file downloaded, if any, and a date section 772 provides date information concerning when the file was downloaded.

Status section 768, as shown, provides various types of status information, such as the following: a ready status indicates that the selected file is ready to be downloaded; a downloaded status indicates that the selected file has been successfully downloaded; an in progress status indicates that the selected file is currently being downloaded with completion column 770 providing an indication of the percentage downloaded; a cancelled status indicates that a request to download the selected file has been cancelled. A details section 774 provides information concerning a particular selected file in section 764. The highlighting indicates that a file in section 764 has been selected by a user, and the user may select the file in the same manner that a user selects a section.

In downloads screen 750, selection of an open/install section 754 (step 574) provides for the user to request installation of a particular file as selected in status section 764. In response, the agent determines if the requested file has been downloaded (step 586). If the file has not been downloaded, the agent may provide an error message to the user indicating that the file

must first be downloaded before it can be installed (step 588). If the file has been successfully downloaded, the agent installs the file (step 590). Installation of the file typically requires any conventional installation permitting for execution of the file by user machine 201. Prior to installation, the agent may display an install screen 776, shown in FIG. 18, requesting that the user confirm whether to install the selected file or not through a selection of section 777 to confirm installation or selection of a section 778 cancel installation.

Selection of a scheduled section 756 in downloads screen 750 permits the user to schedule downloading of a file selected in status section 764. Selection of schedule section 756 (step 576) causes display of a schedule download screen 780 shown in FIG. 19 (step 592). Using schedule download screen 780, the agent receives and stores schedule information for the selected file for downloading a selected file (step 594). Schedule information includes any type of date-based or time-based information for use in scheduling downloading of files or searching for file updates. For example, a user may enter a date in date section 781 and a time in time section 782 in order to schedule a download for the selected file. By selecting section 783 the user confirms the schedule and by selecting a section 784 the user cancels the function.

Selection of a remove section 758 in downloads screen 750 causes removal of a selected file in status section 764. Upon selection of remove section 758 (step 577), the agent removes the file indication from status section 764 if verified by the user (step 593).

Selection of a help section 762 in downloads screen 750 (step 578) causes display of a help screen 786 shown in FIG. 20 (step 596). Downloads help screen 786 permits display of information concerning downloading of information to a user. It typically includes connect section 730 in order for the user to connect with the web site and obtain more help information.

The agent determines if the user selects connect section 730 (step 598); if so, the agent launches the web browser and connects with the appropriate web site to obtain more help information (step 600).

In downloads screen 750, selection of an updates section 720 (step 580) causes execution of an updates screen routine (step 546), as further explained below. Selection of home section 702 (step 582) causes the agent to return to display home screen 700 and execute home screen routine 500.

FIG. 7 is a flow chart of downloads routine 584 executed upon a user's selection of download section 752 in downloads screen 750 or based on schedule information as determined in steps 557 and 558. Upon selection of download section 752, the agent displays a download status screen 788 as shown in FIG. 21A. In downloads routine 584, server 205 typically transmits a file list to the agent (step 606). The file list provides for the identification of files in status section 764. The list may be transmitted one time or repeatedly transmitted and updated. The user selects a file in the file list download (step 608). Alternatively, the user may have preselected a file for downloading using schedule information. The agent sends a get file request to the server, including a URL and starting point of where to begin transmission of the file (step 610). In this example, the starting point is a start byte; however, the starting point may include any type of indication of where to begin transmission of the file. The agent may optionally encrypt the request using any type of encryption routine.

If this was the first time the file is requested to be downloaded, the start byte will be zero. By tracking start bytes, the agent may maintain an indication of an amount of the file downloaded and therefore need only request to download an additional portion if the download is interrupted.

In response to the get file request, server 205 downloads the requested file to the agent by serial transmission of digital information for the file using conventional Transmission Control Protocol/Internet Protocol (TCP/IP) transmission (step 612). In this example, the serial transmission occurs through a stream of bytes for the file; however, it may occur through transmission of other forms of digital information such as, for example, bits, contiguous collections of bytes, or packets. The transmission may occur through any wireline or wireless network.

As the agent receives the file, it stores the file in a temporary folder on the user's machine such as "C:\temp\" on the hard disk drive for the user machine, and it uses a counter to track receives bytes. It also appends the bytes to a previously-received portion, if present in the temporary drive, for downloading the file in multiple portions (step 614). It may store the bytes under a file name related to the file in order to detect a previously-received portion under the same name, if present. It may simultaneously store and append the bytes, or separately perform those steps. Also, instead of using the temporary drive, the agent may store the file in other drives or storage mediums.

Table 1 provides the levels of protocol used for the transmission.

Table 1

level	protocol
1	Agent Transfer Protocol (ATP)
2	encryption
3	TCP/IP

The ATP (level 1) includes the processing shown in downloads routine 584 (FIG. 7). The encryption protocol (level 2) provides for optional encryption of the transmitted bytes using various types of encryption such as triple Data Encryption Standard, referred to as 3DES. Level 3 includes the standard TCP/IP transmission of data over the Internet.

5 While the file is being downloaded, the agent displays and continually updates download status screen 788 (step 616). Download status screen 788, in particular, includes a file identification section 789 providing for identification of the file being downloaded, a status section 790 providing for an indication of the status of the file, whether a new download or a continued download, and a section 791 providing for additional status information such as a size of the file and an estimated remaining time to completion of the download. Section 791 also provides a visual indication of download in progress by continuously displaying the current number of bytes downloaded along with the total number to be downloaded. A section 792 provides for another visual indication of the download as in progress. In particular, it provides for an expanding status bar displaying essentially in realtime a relative indication of the amount that the file is downloaded. The status bar moves, for example, from left to right filling up section 792 until the file is completely downloaded. Selection of a cancel section 793 permits the user to cancel the download.

 The agent may track a percentage or a relative amount downloaded by tracking received bytes and by knowing in advance a file size corresponding with the file. For example,
20 conventional operating systems such as the Microsoft Windows program includes the ability to store file sizes in bytes associated with file names and potentially other information associated with the file such as a type of file and when the file was last modified. The agent continuously

divides the received number of bytes by the file size in bytes to display the status bar and the percent downloaded.

Interruption of the transmission may occur in a number of ways such as, for example, through a user's selection of cancel section 793 in download status screen 788, through any type of loss of the network connection, or through software or web browser failure. For example, transmission interruption may occur if a user turns off power to the user machine, if the user machine otherwise experiences a loss of power, if the user disconnects it from the network, if a server or router for network transmission fails, or through any other type of software or hardware failure.

If the transmission was interrupted (step 618), the agent requests whether the user wishes to continue downloading (step 620). For example, the agent may display a continue download screen 794 as shown in FIG. 21B. Continue download screen 794 includes a section 796 for the user to select in order to continue the download and a section 798 for the user to select in order to not continue the download at this time. If the agent continues to download the same file at another time, it may search for the same file name in the temporary folder, along with the associated start byte number, in order to download only the remaining portion and append it to the previously-received portion before the interruption occurred. Continue download screen 794 may alternatively include another option to cancel the download and delete the portion of the file already received.

If the user wants to continue transmission (step 622), the agent sends another get file request including a URL of the file and start byte (step 610). Since the agent tracks received bytes, it may now continue transmission by the next start byte following the last byte received. In

particular, the agent sets the starting byte to the number of the last byte successfully received plus one. The download process then continues in steps 612, 614, and 616, and the agent appends the remaining bytes to the previously-received portion of the file in the temporary folder, for example. If the downloading is not interrupted (step 618), the agent monitors the download to determine if it receives an end of file (EOF) indication (step 626); downloading continues while uninterrupted and until an EOF indication is received. The EOF may include any type of symbol, character, code, or other information providing an indication that t

Once the agent receives the EOF indication, it has successfully downloaded the file, and it requests whether the user wishes to install the file (step 628). The agent may, for example, display installation screen 776 (FIG. 18) in order to request installation. If the user does not request installation, the agent stores the file on the desktop (step 632), and the user may, at a later time, request installation. Otherwise, if installation is requested, the user installs the file by performing the necessary processing so that the file is executable by the user (step 630).

Throughout the downloads routine, the agent saves and updates the status in download status section 764 in downloads screen 750 for the particular file being processed (step 624). Updating the status provides for changing the status in status column 768 for the corresponding file in order to indicate its status, as well as providing an indication of the amount downloaded in completion column 770 and an indication of the date when last downloaded in date column 772.

FIG. 8 is a flow chart of updates screen routine 546 executed upon a user's selection of updates section 720 in home screen 700 or downloads screen 750. In routine 546, the agent displays an updates screen 795, shown in FIG. 22, upon a user's selection of updates section 720 (step 634). Within updates screen 795, the user selects a particular function (step 636) and the

agent in response executes the requested function. Updates screen 795 includes an update status section 802 providing for information concerning updates to particular files. In particular, a name column 803 provides an identification of a file name and, as shown indented, an indication of updates to the corresponding files. A price column 804 indicates a price for the particular update, and a type column 805 indicates that type of update such as an add-on. A details section 806 provides for information concerning the particular highlighted (selected) update in update status section 802.

In updates screen routine 546, selection of download section 797 (step 638) causes execution of downloads routine 584 (FIG. 7) for the selected update (step 650). A selected update is shown with highlighting, for example, and a user may select an update in the same manner as selecting a section. Selection of a find updates section 799 (step 640) causes execution of a find updates routine as further explained below (step 652). Selection of a remove section 801 (step 642) causes the agent to remove a file or update selected by user, if the user verifies the removal (step 654). Selection of a help section 807 (step 644) causes display of a help screen 808 shown in FIG. 23 (step 656). Help screen 808 provides for display of information to the user concerning finding updates for files. Help screen 808 specifically includes connect section 730 for a user to connect with an appropriate web site in order to obtain more help information. From help screen 808 the agent determines if the user selects connect section 730 (step 658); if so, the agent launches the web browser and connects with the appropriate web site in order to obtain more help information (step 660).

Selection of downloads section 718 in updates screen 795 (step 646) causes the agent to execute downloads screen routine 544 (FIG. 6). Selection of home section 702 (step 648) causes the agent to display home screen 700 and execute home screen routine 500 (FIGS. 5a and 5b).

FIGS. 9a and 9b are a flow chart of find updates routine 652 executed upon selection of find updates section 799 in updates screen 795 or based upon schedule information. Find updates routine 652 permits the agent through interaction with server 205 to automatically search for updates to particular files. The agent and server use application signatures in order to uniquely identify the files stored on a user system. By uniquely identifying those files, the server may determine whether updates exist for particular files. The use of application signatures is particularly useful when files have many different versions, as often exists for software applications, and the server must determine whether an update exists for a particular version of the file on a user machine. The term "application signature" refers to any information used to uniquely identify a file, an example of which is provided below.

In find updates routine 652, the user selects find update section 799 (step 666). The agent may include an end user license agreement for the user. If it has an end user agreement, the agent determines if this is the first time the user has requested to find updates (step 668); if so, the agent displays an end user license agreement (step 670) and determines if the user accepts the agreement (step 672). The agent may be configured to only look for updates if the user has accepted the end user license agreement. Alternatively, the agent may operate without an end user license agreement or with varying versions of such agreement.

Based upon a user's selection of find updates section 799 or upon schedule information as determined in steps 557 and 558, the agent searches for updates. The agent scans the hard disk

drive on the user's machine in order to search for executable (.exe) files, dynamic link library (.dll) files, and driver (.drv) files, and the agent constructs a message including the directories and file names (step 674). Executable files are identified by an ".exe" suffix, dynamic link library files are identified by a ".dll" suffix, and driver files are identified by a ".drv" suffix. The agent may alternatively look for other types of files; however, in the exemplary embodiment it generally searches for the executable, dynamic link library, and driver files. The agent returns a get file information message to server 205 with application signatures (step 676).

In this example, the application signatures include a file name and corresponding size in bytes. Certain windows applications, for example, provide file name and size information, and the agent may obtain the file name and size information from such an application or generate the information itself. For certain files, different versions of the executable file may have the same size and, therefore, the executable file name plus size will not alone uniquely identify a particular version. In that case, the agent and server may also use corresponding dynamic link library file names plus associated sizes to construct the application signatures, and for that signature to match, both the executable file name and size, and dynamic link library file name and size, must match a stored signature. Driver files may be used in the application signature in the same manner, requiring a match between driver file names and sizes in addition to the matches between other types of file names and sizes. Therefore, an application signature in this example may include multiple file names and associated sizes for a particular file. Various types of files associated with a single file may be used to generate a unique application signature, depending upon application signatures already stored in the server database.

Server 205 uses the application signature to determine if updates exist for the corresponding files (step 678). In particular, the server selects an application signature (step 680) and determines if a match exists in an updates database by individually comparing the received file names and associated sizes with stored file names and associated sizes (step 682). If a match exists, the server retrieves the corresponding update information (step 684). Update information includes any type of indication of updates to files. The server may include links to a different database including the update information based upon the application signature. The server then determines if more application signatures exist to be processed (step 686); if so, it returns to step 680 to process additional application signatures. After processing all of the application signatures, the server sends to the agent a file information response with recommended updates from the retrieved update information (step 688). The server also checks if the update recommendation is on the user's machine; if so, it skips the update.

The agent presents an indication of the recommended updates to the user, for example, as shown in update status section 802 in updates status screen 795 (step 689). The agent determines if the user selects an update to be downloaded (step 690). For example, user may select an update and then select download section 797 in order to download that update. If the user requests to download an update, the agent executes the download routine 584 (FIG. 7) to download the selected update (step 692).

While the agent executes processing to find the updates, it displays update information in an update status screen 810 shown in FIG. 24 (step 693). In particular, update status screen 810 includes a section 811 for the agent to indicate status of preparing to search, a section 812 indicating that it is scanning the user system to obtain particular hardware information and file

information, a section 813 to indicate that it is checking for updates with the server, and a section 814 indicating that it has received update information from the server and is compiling the results. In this example, the status of each section is indicated by a check-mark displayed in the corresponding box. A cancel section 815 permits the user to cancel the selected function.

5 In find updates routine 652, steps 678, 680, 682, 684, 686, and 688 are executed by the server, while the agent executes the other steps.

While the present invention has been described in connection with an exemplary embodiment, it will be understood that many modifications will be readily apparent to those skilled in the art, and this application is intended to cover any adaptations or variations thereof. For example, different labels for the various modules and databases, and various hardware embodiments for the servers and machines, may be used without departing from the scope of the invention. This invention should be limited only by the claims and equivalents thereof.